Pega 7.1.8 Installation Guide

Database: Oracle

Application Server: WebSphere

BETTER BUSINESS SOFTWARE®
FOR THE DIGITAL ENTERPRISE

Pega®

Copyright and trademark notices

Copyright

Trademarks

Consult the 718thirdpartylicense.pdf file on the installation media for information about the software delivered with the product.

Notices

This information is the property of:

Pegasystems

One Rogers St.

Cambridge, MA 02142-1209

USA

Phone: (617) 374-9600

Fax: (617) 374-9620

www.pega.com

Updated: Tuesday, July 07, 2015

Contents

Using Pegasystems documentation

This guide describes how to install a new instance of Pega 7.1.8 on a system with Oracle and WebSphere. See the *Platform Support Guide* on the PDN for a list of supported platforms.

To upgrade, see the *Pega 7.1.8 Upgrade Guide*. To update from Pega 7.1.x, see the *Pega 7.1.8 Update Guide*.

Online resources

The Pega Discovery Network (PDN) at https://pdn.pega.com is Pegasystems' online documentation and information site. To access the latest documentation, navigate to the **Support & Resources** tab.

- ***Platform Support Guide***: The *Platform Support Guide* lists the databases, drivers and application servers supported for this release. Review the *Platform Support Guide* before you install Pega 7 to verify that your database and application servers are supported.
- **Deployment guides**: The PDN includes the latest installation, upgrade, and update guides.
- **Release notes**: Review the important information in the release notes before you continue.
- **Updated help files**: Download the current prhelp.war file from the **Resources & Support** tab of the PDN, or use the online version: https://pdn.pega.com/sites/pdn.pega.com/files/help_v71/procomhelpmain.htm.

Installation Overview

Installing Pega 7 is a multi-step process that involves configuring your database and application server, loading rules into the database, and then deploying application archives to the application server.

This guide is organized to expedite the installation process. It can be summarized in the following steps that are typically performed in sequence.

1. Configure the database
2. Configure the application server
3. Install Pega 7
4. Configure Pega 7

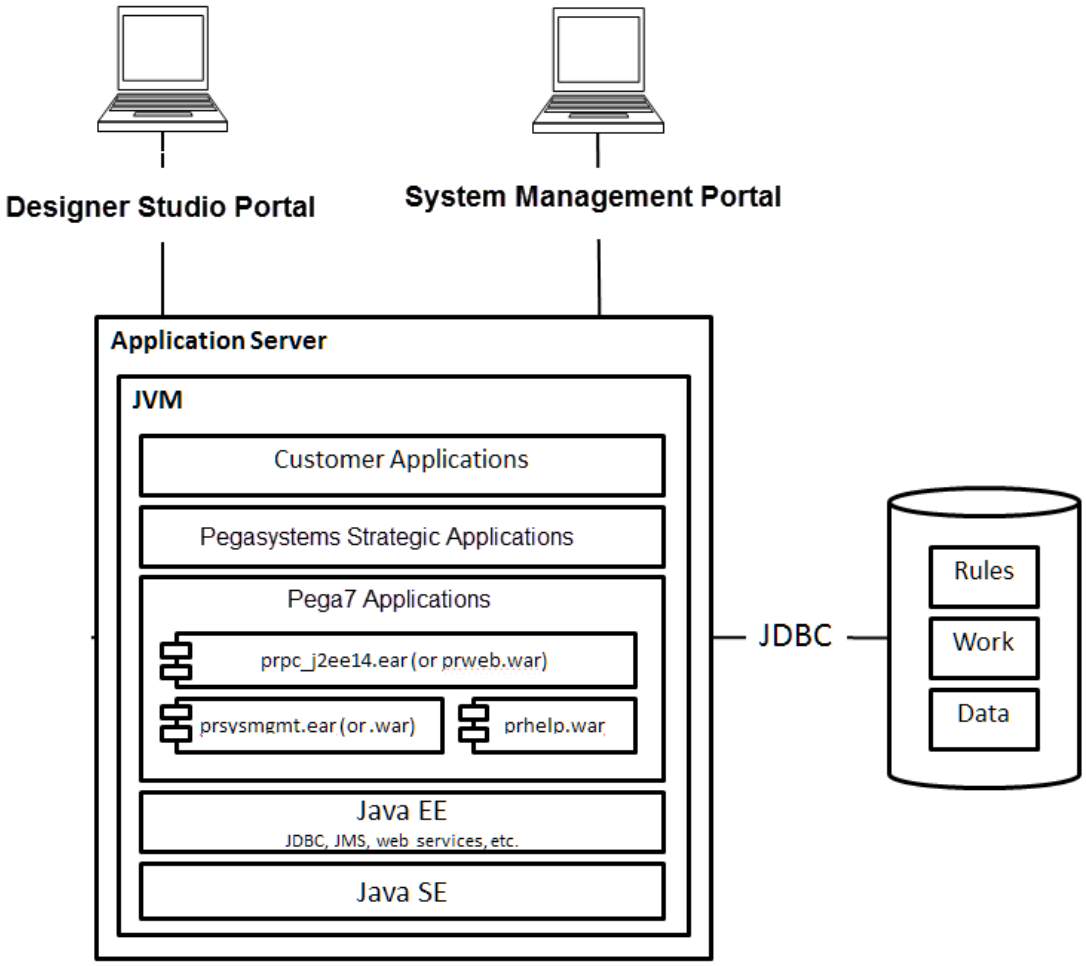
Some organizations follow specific governance procedures for the allocation and use of IT infrastructure, including databases and application servers. If your organization has special teams for managing databases and application servers, these resources should be engaged as early in the planning process as possible. Pega 7 supports different deployment topologies and configuration options which impact how the supporting infrastructure is configured and managed going forward.

Pega 7 architecture

Pega 7 is a Java EE-compliant enterprise application that requires two main subsystems for its operation — an application server and a database server.

- The **application server** hosts Pega 7's application archives and also provides interconnectivity to other systems through various protocols.
- The **database server** stores the rules, data and work objects used and generated by Pega 7.

Application users and developers typically access Pega 7 through a web browser. Applications may also expose HTTP-based services (for example, SOAP, REST or HTTP) for purposes of administration or process automation in a headless environment.



Planning your installation

Pega 7 supports a run-time mode and a variety of deployment topologies that can impact the choices you make at different stages of the installation . Before beginning, read this section thoroughly.

Note: If you are making changes in your environment as part of the upgrade, do them before upgrading.

Single and split-schema systems

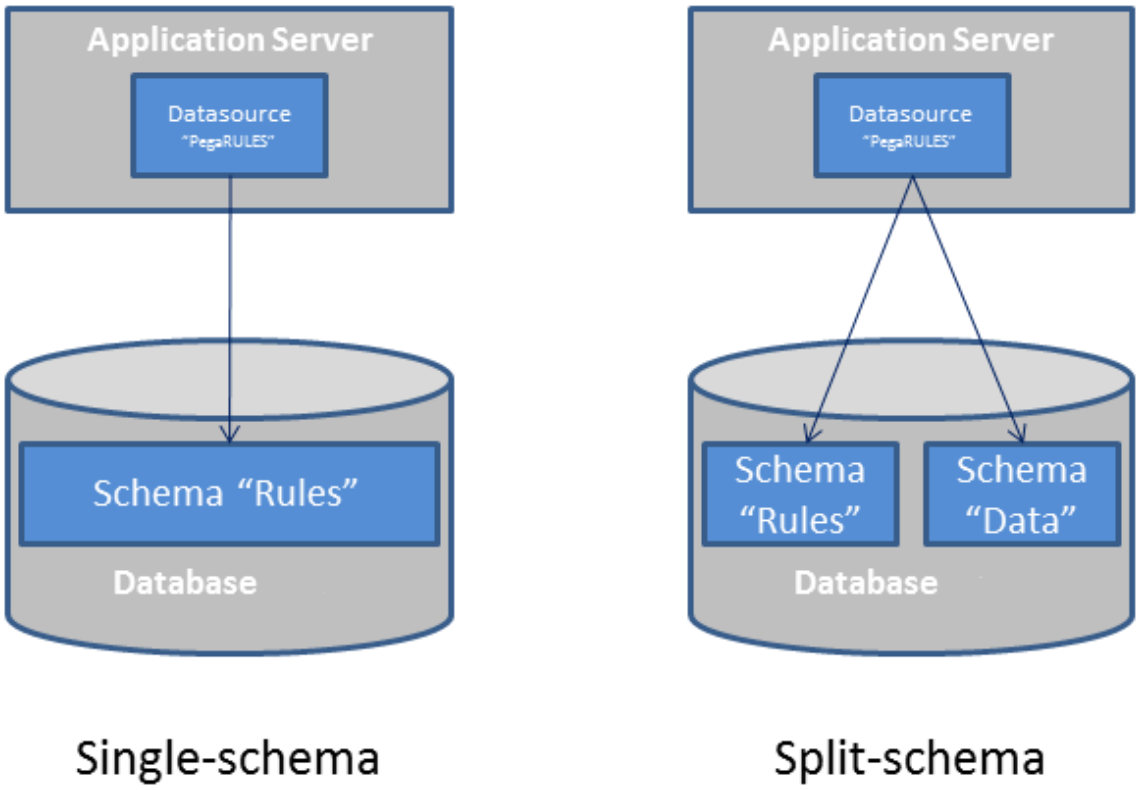
There are two configuration types for Pega 7:

- **Single schema**: One schema contains all rules and data objects.
- **Split schema**: There are two schemas on the same database:

- a **Rules** Schema – contains Rules tables and associated data and objects
- a **Data** Schema – contains transaction data, including Work objects

Split-schemas allow you to conduct an in-place upgrade in one environment and then migrate the upgraded objects to other environments. Split-schema configurations are best practices, particularly in critical development environments such as QA, STAGE and PROD.

The following diagram illustrates the difference between a single-schema system and a split-schema system.



Installation methods

You can install Pega 7 either automatically or manually. This guide includes instructions for both methods.

- **Automatically** – Use the UI-based Installation and Upgrade Assistant (IUA) to automatically install either the rulebase or the rulebase and the schema. The IUA runs on any Windows, Unix, Linux, or zLinux system with Java 6 or higher.
- **Manually** – Configure the **setupDatabase.properties** file with your site-specific properties.

Then run scripts to install and optionally generate DDL. If you generate DDL, have your Database Administrator apply the DDL to your schema.

Both methods use a batch process to load the rulebase. Due to the magnitude of rules and other data objects which must be loaded, Pegasystems strongly encourages you to run the installer on the same network as the database server. If this is not possible, run the installer on a machine with fast direct access to the database server. Do not attempt to run the installer on a virtual private network (VPN) or a multi-hop wide area network (WAN).

Product mode

Pega 7 includes a Multitenancy mode. The Multitenancy mode has different requirements, different run-time behaviors and different administrative procedures from other modes. Before you select the Multitenancy mode, review the *Multitenancy Administration Guide* which discusses this mode in detail. It is included in your distribution package and available on the PDN.

Transport-layer encryption

Pegasystems recommends that you use a strong transport-layer encryption (for example, TLS1.2) to secure Pega 7 web applications. This requires that you create and install TLS/SSL digital certificates on your web application server for Pega 7. For more information, see the documentation for your application server.

Business Intelligence Exchange (BIX) and Pega 7 product versions

Release versions of the Business Intelligence Exchange (BIX) product are synchronized with release versions of Pega 7. To use the BIX product once you have obtained a license, the version of BIX must be the same as the version of Pega 7. BIX is included in the full distribution for Pega 7.1.8. See the Pega 7.1.8 BIX User Guide for information about installing BIX.

Triggers in Pega 7.1.8

Pega 7.1.8 handles triggers differently than previous versions. Triggers that reference the updates cache and rule view tables will drop when Pega 7.1.8 opens for the first time. Pega 7.1.8 now maintains these tables. If you have custom triggers that only maintain these tables, do not recreate the trigger. If you have custom triggers that depend on these tables or are defined on these tables and these triggers also have other logic you want to keep, contact Pegasystems Global Customer Support for instructions on how to proceed.

System requirements

Before you install, check the supported versions of your application server and database. See the Platform Support Guide on the PDN for a list of supported versions. Ensure that your system meets the minimal requirements describes in this section.

Installation and Upgrade Assistant (IUA)

The minimum systems requirements to run the Installation and Upgrade Assistant (IUA) are:

- Windows or Linux operating system
- 1.25 GB minimum available memory
- 10 GB minimum disk space and at least 8 GB available space in the temporary directory of the root file system
- Java Platform, Standard Edition Development Kit (JDK). Set JAVA_HOME to the root directory of the JDK. For Java 8, you may need to remove from the PATH any references to a Java shortcut.

Application server

- Oracle JDBC type 4 driver, such as *ojdbc6.jar*. See the *Platform Support Guide* for more information about supported drivers.
- 10 GB minimum free disk space

  Note: Allocate enough storage to accommodate debugging and other logging requirements. Configure logging to avoid writing logs to the directory that contains the application server run-time components.

Application server memory requirements

Pega 7 runs in memory on Java Virtual Machines (JVMs). This memory is called a "heap". In general, all activity is distributed over multiple JVMs (nodes) on the application server.

- Standard suggested development system heap size is 4GB.
- Standard suggested production heap size is 8GB.
- Larger heaps may be advisable if a system is to run applications designed to allow a high number of concurrent open tasks per session or to cache a large collection of transaction or reference data.

Oracle JDKs use a compression option to minimize the cost of large heaps. The option is labeled CompressedOOPS. Compression is effective up to 32 GB.

In current 64 bit JVMs, compression is enabled by default. Do not deploy Pega 7 in an environment where compression is disabled or where the heap size exceeds the vendor-specific effectiveness limit.

The host application server memory size must be at least 4 GB larger than the Pega 7 heap size. The additional 4GB is recommended to allow for Java native memory (stack space, classes, class loaders, and so on), operating system and monitoring tools, and operating system network and file buffering.

The minimum host application server memory size is 8 GB:

4 GB heap + 4 GB for native memory, operating system, and buffering

Database Server Setup

This section lists the database prerequisites and describes how to create user accounts, the database, and database schema for your installation.

Preparing your database

Before you begin preparing your database, verify that the database server is installed and running a supported version of the database. Modify your Pega 7 database configuration to match the latest requirements. In particular, pay attention to the requirements described below.

CAUTION: If your system includes synonyms to Pega-supplied tables, drop the synonyms before you upgrade. If necessary, reapply the synonyms after the upgrade is complete.

- Download a supported version of the JDBC4 driver for your version of the database.
- Enable Java in the database if you plan to use User-Defined Functions.
- Verify that the system includes:
    - 16 GB minimum RAM.
    - 10 GB minimum initial tablespace set to auto extend for the Rules user.

      Note: The Rules user and the Data user can share the same tablespace. If you create separate tablespaces for the Rules user and the Data user, base the size of the Data user tablespace on the estimated number of work objects in the application.

    - 50 MB logfile size – this default size is sufficient for the initial installation, but will need to be resized to run the application server workload.
- If you are using Oracle 11g, do **not** use the UCP (Universal Connection Pool) feature in your database. An Oracle bug (BUG 8462305) causes a failure when an application tries to call a stored procedure causing Pega 7 to work incorrectly with a database that uses UCP.

  Note: To determine if UCP is in use, check for the ucp.jar file on the classpath of the application server.

Configure time zone, character encoding and locales

Configure your database server, application server, and the system on which you are running the installation to use the same:

- Time zone
- Character encoding (UNICODE or EBCDIC)

- Regional settings/locale

Enable support for stored procedures

Pega 7.1.8 uses native stored procedures, not external stored procedures. Make sure your site supports SQL-based stored procedures.

Enable support for User-Defined Functions

Enable support for User-Defined Functions (UDF) installed by Pega 7.

Configuring your Oracle database

To prepare your Oracle database server for use with Pega 7, complete the following steps:

1. Create database user accounts.
2. Create an empty database.
3. Create the database schema.
4. Create the Oracle tablespaces.

Database users

This guide refers to the following database users:

- Deployment users: these users perform actions only during the installation.
    - **Deployment user**: the user who runs the installation. After the installation, you can remove this user.
    - **Oracle users**: Because Oracle has a one to one relationship between users and schemas, if you have a split-schema system, you must have separate users for the Rules Schema and the Data Schema. The Oracle Rules Schema user may be associated with individual tablespaces or a common tablespace. Pegasystems recommends separate tablespaces for each user in critical SDLC environments. The Oracle Rules Schema user is only used to create the schema and requires unlimited tablespace.
- Run-time users: These users perform actions on Pega 7 after the installation. In a dual user configuration, an **Admin** user is granted full privileges, and a **Base** user is granted a smaller subset. Pegasystems recommends the dual user configuration :
    - **Base user**: the user who will run Pega 7. For most run-time operations, Pega 7 uses the Base user and associated data source. The Data Schema should be the default schema for Base users.
    - **Admin user**: an optional user provided to Pega 7 that is preferentially used by certain features that require creating, modifying, or dropping database schema objects; for example, System Management facilities and certain decisioning operations.

Pegasystems recommends that you create the Admin user separate from the Base user, however, you can create a single Base user with both sets of privileges. If there is no separate Admin user, Pega 7 uses the Base user for all runtime operations.

Creating Oracle Users

Use either a SQL command or the Oracle Enterprise Manager to create users with these privileges and roles. Because Oracle maintains a 1:1 relationship between schemas and database users, this process also creates the schemas.

Deployment user privileges and roles

The Deployment user requires these privileges and roles for all configurations:

- CREATE SESSION
- CREATE ANY TABLE
- ALTER ANY TABLE
- INSERT ANY TABLE WITH ADMIN OPTION
- SELECT ANY TABLE
- UPDATE ANY TABLE
- DELETE ANY TABLE
- CREATE ANY INDEX
- CREATE ANY PROCEDURE
- EXECUTE ANY PROCEDURE
- CREATE ANY VIEW
- CREATE ANY TYPE
- CREATE ANY TRIGGER
- ALTER ANY TRIGGER
- GRANT ANY OBJECT PRIVILEGE
- DROP ANY PROCEDURE
- DROP ANY TRIGGER
- DROP ANY TABLE

- DROP ANY VIEW
- DROP ANY INDEX
- ANALYZE ANY
- ANALYZE ANY DICTIONARY
- SELECT_CATALOG_ROLE (This is a role, not a privilege.)

  NOTE: If you plan to install a Pega-supplied application on top of Pega 7.1.8, you must grant SELECT_CATALOG_ROLE to the Deployment or Admin user. Some Pega-supplied applications use triggers and the user will need SELECT_CATALOG_ROLE to drop triggers that read from the updates cache and rule view tables.

  If you have custom triggers that read to the update cache or rule view tables, the deployment automatically drops the custom triggers. Manually recreate custom triggers after you deploy Pega 7.1.8.

Oracle schema users

The Oracle schema users require only unlimited tablespace.

Run-time users privileges and roles

The run-time users require different permissions depending on whether or not you have a dual-user configuration.

Dual-user configuration: Admin and Base users

In a dual-user configuration, grant these privileges and roles:

- Admin user:
  - CREATE SESSION
  - CREATE ANY TABLE
  - ALTER ANY TABLE
  - INSERT ANY TABLE WITH ADMIN OPTION
  - SELECT ANY TABLE
  - UPDATE ANY TABLE
  - DELETE ANY TABLE

- CREATE ANY INDEX
- CREATE ANY PROCEDURE
- EXECUTE ANY PROCEDURE
- CREATE ANY VIEW
- CREATE ANY TYPE
- CREATE ANY TRIGGER
- ALTER ANY TRIGGER
- GRANT ANY OBJECT PRIVILEGE
- DROP ANY PROCEDURE
- DROP ANY TRIGGER
- DROP ANY TABLE
- DROP ANY VIEW
- DROP ANY INDEX
- ANALYZE ANY
- ANALYZE ANY DICTIONARY
- SELECT_CATALOG_ROLE (This is a role, not a privilege.)

  NOTE: If you plan to install a Pega-supplied application on top of Pega 7.1.8, you must grant SELECT_CATALOG_ROLE to the Deployment or Admin user. Some Pega-supplied applications use triggers and the user will need SELECT_CATALOG_ROLE to drop triggers that read from the updates cache and rule view tables.

  If you have custom triggers that read to the update cache or rule view tables, the deployment automatically drops the custom triggers. Manually recreate custom triggers after you deploy Pega 7.1.8.

- Base user: The Base user is the DATA schema user in this configuration.
  - Basic read and write access to data and rules tables including rules resolution.
  - CREATE SESSION

Single-user configuration: Base user only

Pegasystems recommends that you create the Admin user separate from the Base user, however, if you opt for a single Base user, grant these permissions:

- CREATE SESSION
- CREATE TABLE
- CREATE PROCEDURE
- CREATE VIEW
- CREATE TYPE
- CREATE TRIGGER

Creating users from SQL

The following SQL statement allows developers to install and run Pega 7 using the default USERS tablespace with no restrictions or quotas.

```
ALTER USER <user> DEFAULT TABLESPACE USERS QUOTA UNLIMITED ON USERS;
```

Run this statement for:

- Oracle schema users:
  - For single schemas, create one Oracle schema user
  - For split-schemas, create separate Oracle Rules and Data schema users.
- Deployment user
- Base user
- Admin user (for dual-user configurations)

Using Enterprise Manager to create users

To create a user:

1. Log into the **Enterprise Manager** using the URL provided by the Database Configuration Assistant. The URL is usually in the form of

   https://<host>:5501/em

   The Enterprise Manager login screen opens.

2. Enter the User Name and Password for the sys operator. Click **Login**.

   User Name = **sys**

Password = **<password>**

The Enterprise Manager home screen opens.

3. Select **Security > Users**. The Users screen opens.
4. Select **Actions > Create User**. The Create User wizard opens. This provides a guided approach to creating users and assigning privileges.

   On the User Account step, enter the **Name** and **Password**.

   Accept the other defaults.

5. Click  . The Tablespaces step opens.

   If you created a dedicated tablespace, choose that tablespace from the menu.

   Accept the other defaults.

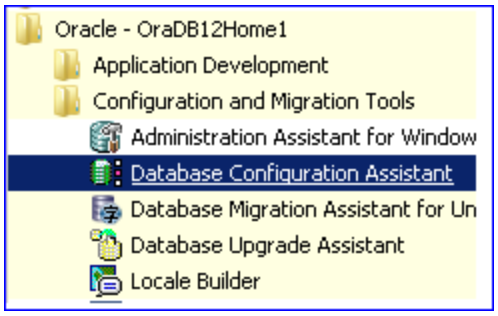
6. Click  . The Privilege step opens.
7. Select the privileges for this user. Click **OK** when you finish adding privileges .
8. Repeat the steps to configure the remaining users.

Creating an empty database

Create a database with a minimum of 5 GB for the user tablespace and, if possible, allow the database to grow. This minimum size allows you to load the initial RuleBase and do simple development. Monitor the database use carefully. As development begins, the size of the database will need to increase significantly, depending on your use of Pega 7 and the number of users.

Using the Database Configuration Assistant to create the database

Use the **Database Configuration Assistant** to create the database. The Assistant provides a guided approach to common database management operations, including database creation and deletion.



1. Select **Create Database** and click **Next>**.

   The **Creation Mode** screen appears.

2. Complete the following fields:

   - Global Database Name
   - Administrative Password
   - Confirm Password
   - Oracle_user Password
   - Uncheck the **Create as Container Database** box.

3. Click **Next>**.

   The Assistant runs **Pre Requisite Checks**. If these checks are successful, the Assistant automatically progresses to the **Summary** screen.

   Optionally copy and paste the summary information to a safe location.

4. Click **Finish** to complete the Assistant and begin database creation.

   Note: It typically takes 5-10 minutes to create the database.

5. When database creation is complete, copy the value of the **EM Database Express URL** to a safe location.

6. Click **Exit** and click **Close** to close the Assistant.

Create Oracle Tablespaces

1. Log into the **Enterprise Manager** using the URL provided by the Database Configuration Assistant. The URL is usually in the form of

   https://<host>:5501/em

   The Enterprise Manager login screen opens.

2. Enter the User Name and Password for the sys operator. Click **Login**.

   User Name = **sys**

   Password = **<password>**

   The Enterprise Manager home screen opens.

3. Select **Storage > Tablespaces**. The Tablespaces screen opens.

4. Select **Actions > Create**. The Tablespace Creation Wizard opens. This provides a guided approach to configuring the tablespace.

   In the Name field, enter the **name** of the tablespace.

5. Click  .

   The Add Datafiles screen opens.

   Increase the File Size to **10G**.

   Accept the other defaults.

6. Click **OK**. The tool automatically generate and executes the SQL.

   Note: If you are installing Pega 7 in a split-schema configuration and wish to maintain Rule and Data Schema in separate tablespace, re-run the Create Tablespace wizard choosing the appropriate values for the Data Tablespace.

   To install Pega 7, configure the Rules tablespace with a 10 GB datafile. The initial Pega 7 rulebase consumes 7+ GB of space. If defined, the Data tablespace should be configured based on a site-dependent analysis of how Pega 7 and any associated Pega-supplied applications or Pega 7 applications will be used in the environment.

Installing Pega 7

This section describes how to install and load Pega 7.

Use one of these methods to install Pega 7:

- **Installation and Upgrade Assistant (IUA)**— A Java-based UI guided tool that sets up the Pega 7 Rulebase schema in the database and loads Pega 7 rules.
- **Command-line script** — A command-line script is provided for both Unix and Windows platforms and is suitable for automating the installation process of installing Pega 7 in headless environments.
- **JCL** – JCL member scripts are provided so that the schema can be generated and applied on the z/OS system itself.

Manual or automatic application of schema changes

As part of installing Pega 7, you must prepare the database schema. You can either use the generateddl script to generate a DDL file of schema changes you can give to your DBA to apply, or the installation process can automatically apply the schema changes.

If you use the generateddl script, edit the setupDatabase.properties file.

Editing the setupDatabase.properties file

Skip this section if your deployment meets both these criteria:

- You will use the IUA to install.
- You will allow the installation process to automatically apply the schema changes.

If your deployment does not meet both these criteria, follow the steps in this section to edit the setupDatabase.properties file. Follow these instructions if you plan to use the command line scripts to install or if you plan to use the generateddl script to generate a DDL file of schema changes. The Deployment user should perform these actions.

The setupDatabase.properties file is used by the scripts that update the database schema and the rulebase. These scripts use the setupDatabase.properties file:

- The **install.bat/sh** script deploys the most recent version of Pega 7.
- The **generateddl.bat/sh** script generates a SQL file to manually apply schema changes.

See Appendix A: setupDatabase.properties for a sample edited file.

Common properties

These properties specify the settings needed to connect to the database:

- The script arguments column lists the arguments for the generateddl.bat/sh script.
- The setupDatabase.properties column lists the corresponding file property.

generateddl script argument	setupDatabase.properties	Description
--dbType	pega.database.type	Database vendor type. Enter: **oracledate**
--dbSchema	rules.schema.name	In a single schema environment sets the Rules Schema and Data Schema. In a split-schema environment, sets the Rules Schema only.
--dbDataSchema	data.schema.name	**Optional**: Specified for a split schema environment only. Sets the Data Schema for the environment.
--mtSystem	multitenant.system	Is this a Multitenant System? true/false
--dbuser	pega.jdbc.username	User name of the Deployment user.
--dbpassword	pega.jdbc.password	Password of the Deployment user. If you plan to use encrypted passwords, leave this blank.
--dburl	pega.jdbc.url	Database jdbc url
--driverJar	pega.jdbc.driver.jar	Path and file name of the jdbc driver
--driverClass	pega.jdbc.driver.class	Class of the jdbc driver
--tempdir	user.temp.dir	**Optional**: the location of the temp directory. If you leave this blank, Pega 7 uses the default. Pegasystems recommends that you set this to C:\TEMP.
--connProperties	jdbc.custom.connection.properties	**Optional**: Semicolon-delimited custom JDBC properties (for example: prop1=value;prop2=value;prop3=value)

Editing the file

To edit the setupDatabase.properties file:

1. Open the file **setupDatabase.properties** for editing. This file is located in the **scripts** folder of your distribution.
2. Specify the properties for your system. For each property, add the appropriate value after the equals sign.
3. Save and close the file.

Manually generating and applying DDL

Skip this section if you do not need to generate a DDL file.

A generateddl script is provided for Windows (**generateddl.bat**) and Linux (**generateddl.sh**) in the **<PRPC_HOME>\scripts** directory. This script automatically renders the platform-specific DDL and writes the output to a file. You can then view and edit the file or directly apply it using native or third-party database management tools. Both scripts work identically and accept the arguments noted in Editing the setupDatabase.properties file.

Use the following convention:

```
generateddl.bat --action  install  --arg1 [value] --arg2 [value] ..... --argn [value]
```

Note: You must supply the **--action** parameter to the script. You can also supply an **--outputDirectory** parameter. If no additional arguments are passed to the script, the script defaults to the values of the properties set in the **setupDatabase.properties** file. See Appendix A: setupDatabase.properties for a sample edited file.

To use the script:

1. Perform these steps as the Deployment user.
2. If you have not already done so, edit the **setupDatabase.properties** file. See Editing the setupDatabase.properties file.
3. Open a command prompt and navigate to the **scripts** directory.
4. Type **generateddl.bat** or **generateddl.sh** passing in the required **--action install** parameter. For example:
    - Single-schema: `#generateddl.bat --action  install  --dbSchema <Rules Schema>`
    - Split-schema: `#generateddl.bat --action  install  --dbSchema <Rules Schema> --dbDataSchema <Data Schema>`

If you do not pass another value for **--outputDirectory**, the script writes the output to the following directory:

<PRPC_HOME>\schema\generated

Note: The output directory is deleted and recreated each time the **generateddl** script executes. To save a copy of the DDL, rename the directory before you run the script.

Applying schema changes

Before you continue, have your DBA apply the schema changes.

Editing setupDatabase.properties to bypass DDL generation

After your DBA applies the changes to your database, set the property **bypass.pega.schema=true** in the setupDatabase.properties file. Then, when you run `install.bat/sh` , the system does not try to apply a schema that is already present. Doing so would cause the installation to fail.

Rulebase Prerequisites

Prior to running the installer — either the IUA or the command-line version — confirm you meet the following prerequisites:

- Database users and roles are defined with the proper privileges
- A new database is defined with sufficient capacity
- Database schemas are defined

  For a **single schema** configuration, you should have one schema for both rules and data objects.

  For a **split schema** configuration, you should have separate schema for rules and data objects.

Configuring Kerberos authentication

Pega 7 supports Kerberos functionality. Kerberos is a computer network authentication protocol which allows nodes communicating over a non-secure network to prove their identity to one another in a secure manner.

To use Kerberos for authentication, you must use the command line to install Pega 7.

To use Kerberos authentication:

1. Edit the setupDatabase.properties file.

   a. In the "Uncomment this property section" of the file, uncomment the **jdbc.custom.connection.properties** property.

   b. Provide the correct parameters to the jdbc.custom.connection.properties property as semicolon-delimited name/value pairs. The specific parameters depend on your security infrastructure:

jdbc.custom.connection.properties=<parameter1>=<value1>;<parameter2>=<value2>;<parameter3>=<value3>;

c. Comment out all the username and password properties so that they appear as follows:

pega.jdbc.username db username

pega.jdbc.password db password

[lines removed here]

pega.jdbc.username=ADMIN

pega.jdbc.password=ADMIN

2. Set up your database to enable Kerberos functionality. This may include additional vendor-specific JDBC driver configuration, or other setup procedures. Check the documentation from your database vendor to determine what Kerberos setup is needed for your database.
3. Continue at Installing from the command line to install Pega 7.

Installation and Upgrade Assistant (IUA)

Because of the large volume of data loaded by the Installation and Upgrade Assistant (IUA), you should run the IUA or command-line scripts on the same network as the database server. If this is not possible, run the tool on a system with fast, direct access to the database server.

The installer creates a series of log files in **<PRPC _HOME>\scripts\logs**. If you encounter a problem while running the installer, copy the logs to a safe location before re-running the installer. The IUA re-creates this directory each time the installer runs.

Note: The **<PRPC_HOME>\.checksum** directory provides an MD5 checksum for each the file in the distribution. To verify that the distribution has been downloaded correctly, calculate a checksum using the Jacksum tool at www.jonelo.de/java/jacksum/.

For example: if your distribution was unzipped to <PRPC_HOME>:

```
java -jar jacksum.jar -m -a md5 -r -p -O outputFile.md5 <PRPC_HOME>
```

Compare **outputFile.md5** to the md5 file located in **<PRPC_HOME>\.checksum**. The checksum values should be identical.

Running the Installation and Upgrade Assistant (IUA)

Before running the IUA, make sure that your system meets the requirements.

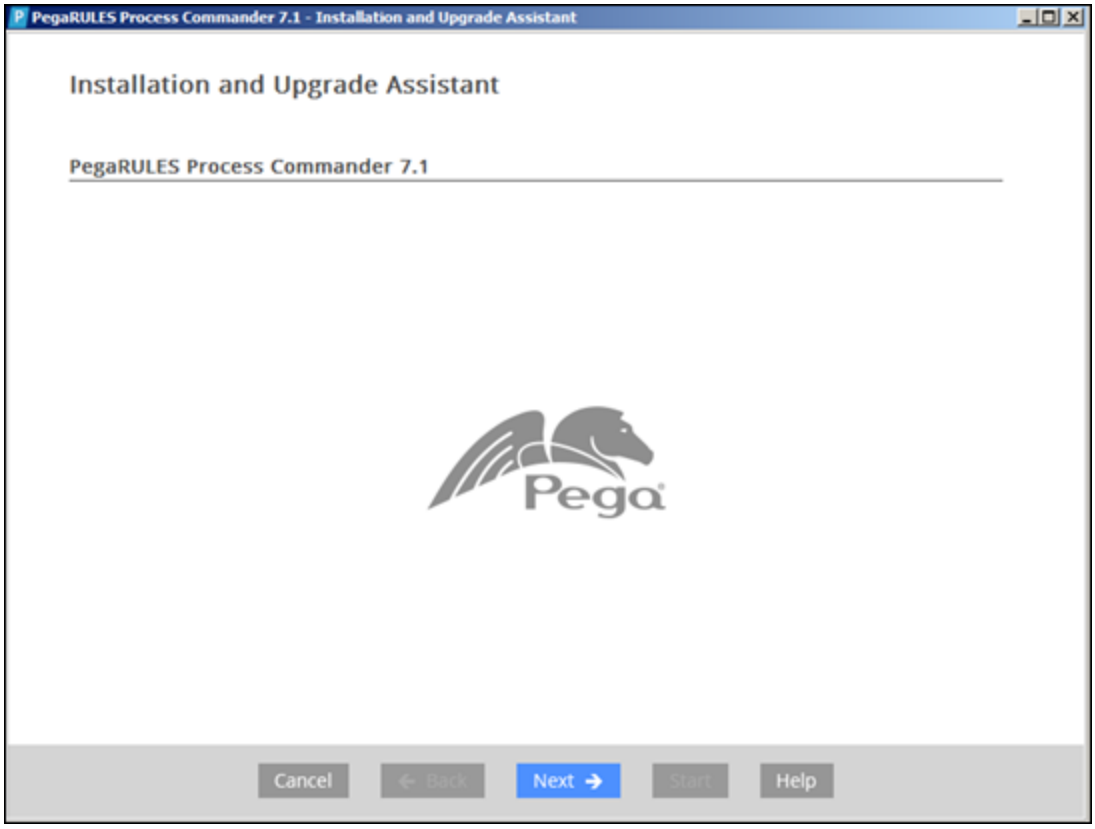
1. Copy the distribution file to the computer that you will use to run the IUA. Extract the contents of the ZIP file into an empty directory.

   If you are installing the software from the DVD included with your media distribution, copy the contents of the DVD to an empty directory.

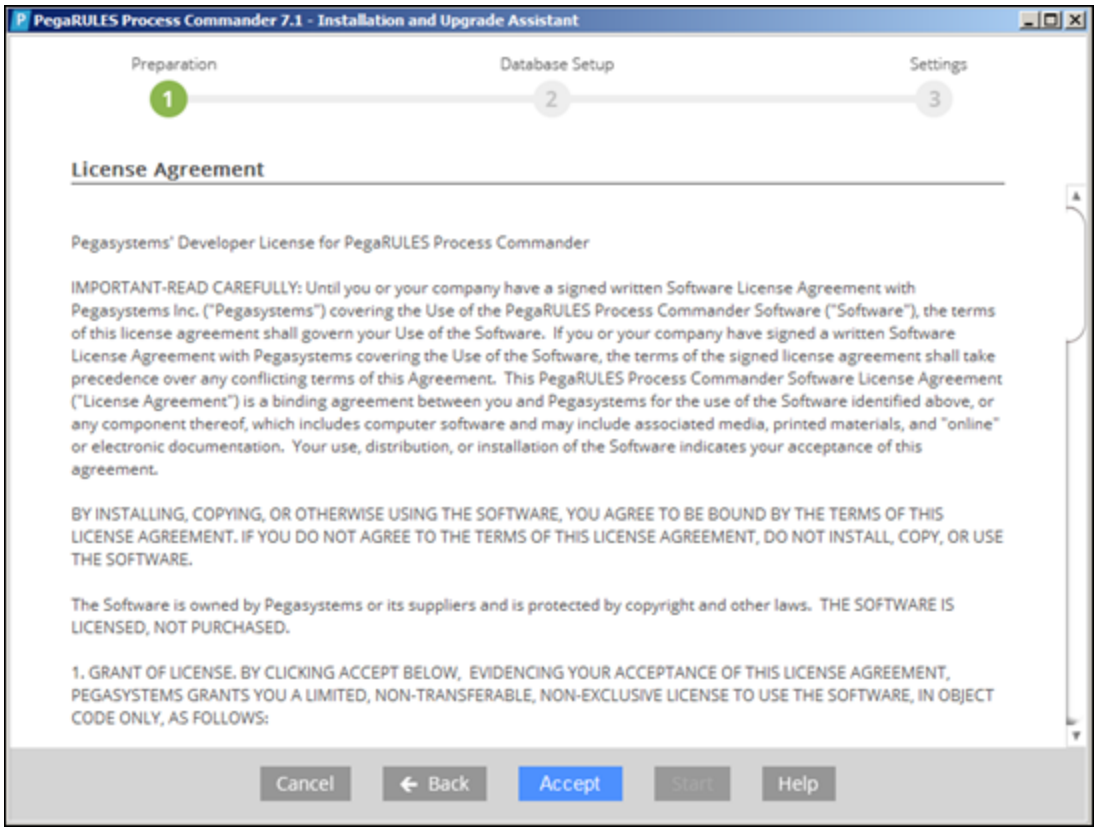
2. Start the IUA; either double-click the **PRPC_Setup.jar** file or run the command **java –jar -Xmx2048m –jar PRPC_Setup.jar**

   Note: If JAR files are not associated with Java commands on your system, start the IUA from the command-line. In a command-line console, change to the directory containing PRPC_Setup.jar, and type **java -Xmx2048m -jar PRPC_Setup.jar**.

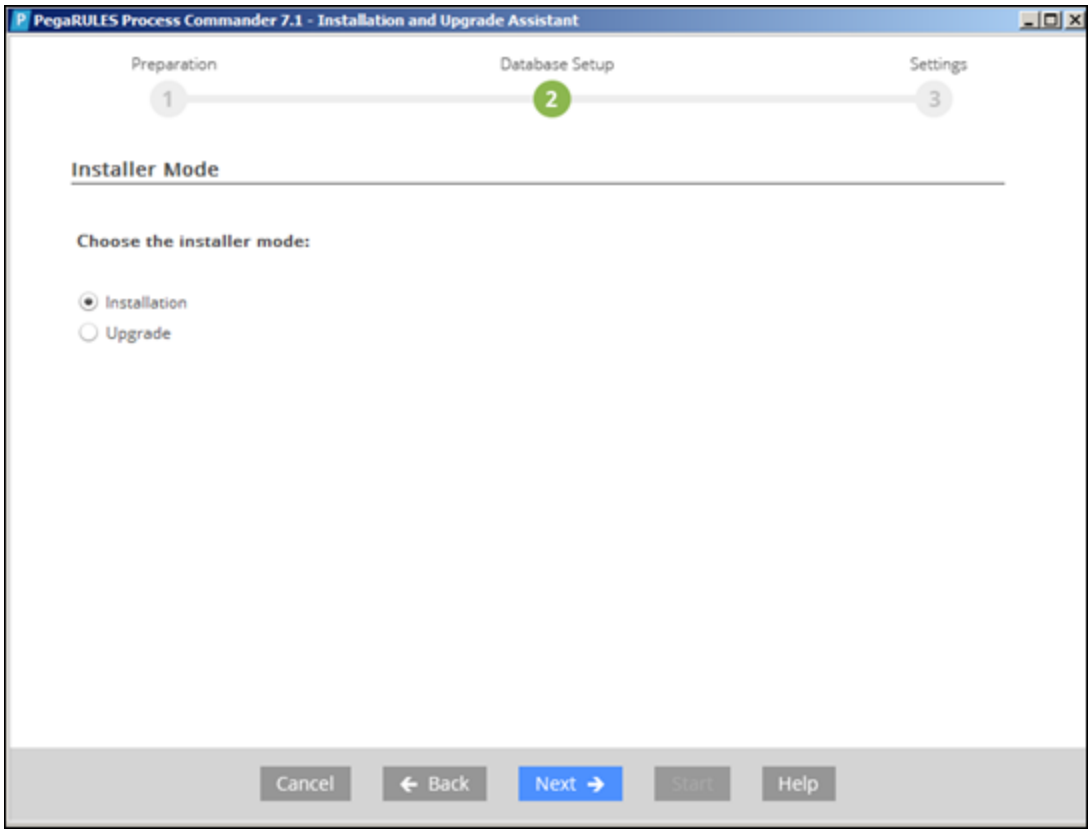
   The IUA loads and the install icon appears in your task bar.



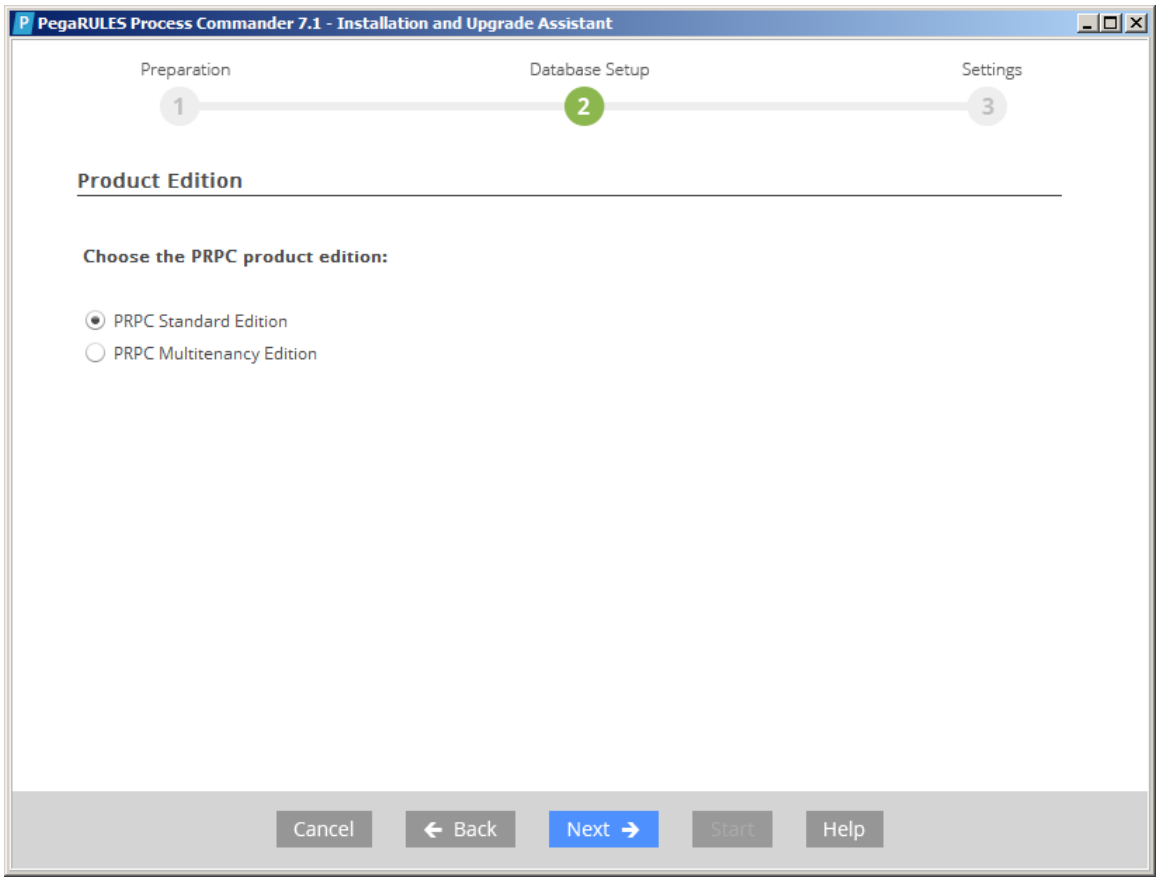
3. Click **Next** to display the License Agreement.



4. Review the License Agreement and click **Accept**.
5. On the Installer Mode screen, choose **Installation** and click **Next**.



6. Choose your Database Type and click **Next**.
7. Choose the product edition and click **Next**.



- **Standard Edition** — this option installs a new instance of Pega 7.1.8. It is the default selection.
- **Multitenancy Edition** — this option installs Pega 7.1.8 in a specialized mode designed to support Software As A Service (SaaS) run-time environments. See the *Multitenancy Administration Guide* on the PDN for more information.

Note: Upgrading or migrating from one edition to another is not supported. If you install one edition and later decided to use a different edition, you will need to drop and recreate the database or create a new database. The schema DDL between the two editions are not compatible.

8. Configure the database connection. The JDBC drivers allow the Pega 7 application to communicate with the database.

Note: Some of the fields on the Database Connection screen – in particular the **JDBC Driver Class Name** and the **Database JDBC URL** - are pre-populated based on the type of database you selected. If you edit these or any other fields on this screen, and then later decide to change the database type, the IUA may not populate the fields correctly. If this occurs, enter the correct field values as documented below, or exit and re-launch the IUA to select the intended database type.

- **JDBC Driver Class Name** — Verify that the pre-populated value is accurate:

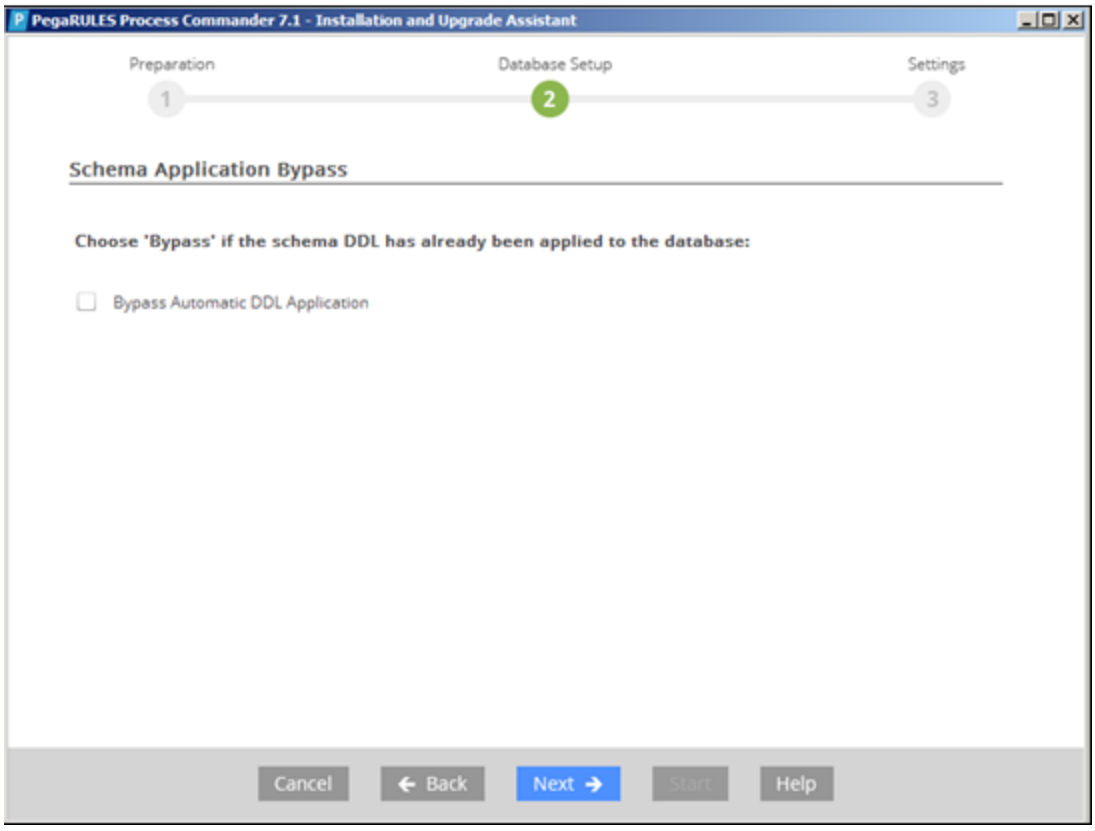
  oracle.jdbc.OracleDriver

- **JDBC Driver JAR Files**— Click **Select Jar** to browse to the appropriate driver files for your database type and version. See the *Platform Support Guide* on the PDN for a list of drivers.
- **Database JDBC URL** — Verify that the pre-populated value is accurate. Replace **servername** and **dbname** with the values for your system:

  jdbc:oracle:thin@//<*serverName*>:1521/<*dbName*>

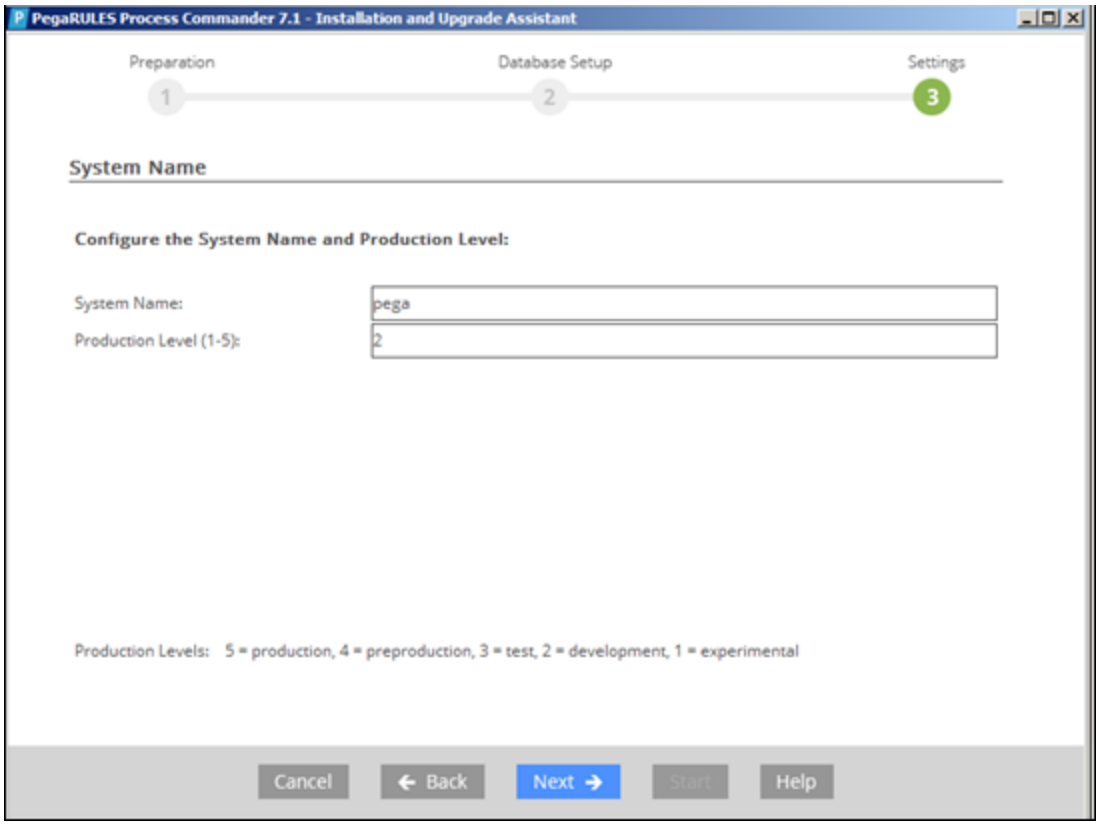
- **Database Username and Password** — Enter the user name and password you created for the Deployment user on your database.
- **Rules Schema Name** — Enter the name of the Rules Schema in the database.
- **Data Schema Name** — Enter the name of the Data Schema in the database. For single-schema configurations the Data Schema Name is identical to the Rules Schema Name.

9. Click **Test Connection**. If the connection is not successful, review your connection information, correct any errors and retest. When the connection is successful, click **Next** to choose how to apply the database schema.



10. Specify whether to bypass automatic DDL generation. By default, the IUA is designed to automatically generate and apply the schema DDL to your database. If you already manually generated the DDL and your Database Administrator has already applied the DDL, select the **Bypass Automatic DDL Application** option. Otherwise, leave the box unchecked. Click **Next**.

11. Enter the System Name and Production Level and click **Next**:

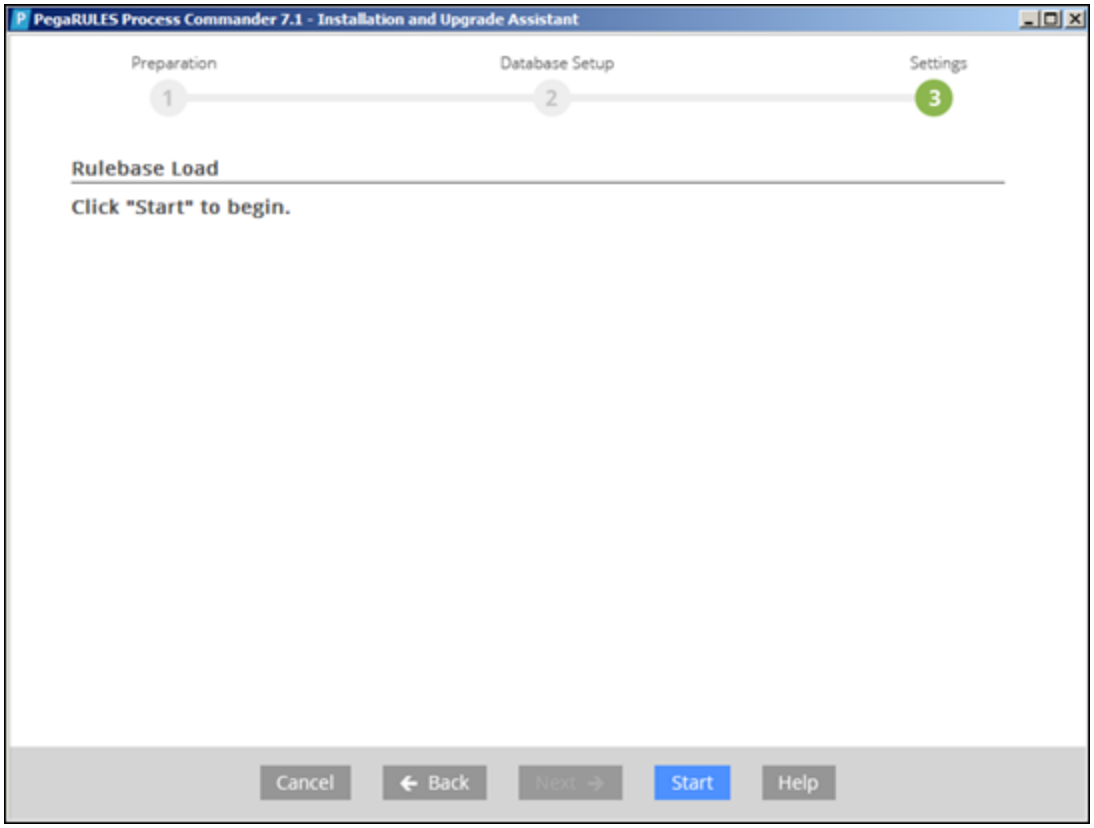


- **System Name** — Enter a name for your Pega 7 system. The System Name can be accessed from the System > Settings > System Name landing page.
- **Production Level** — Enter a production level. The Production Level affects many security features of your system. Both the System Name and Production Level can be changed after the system is running. Depending on the type of installation, choose:
  - **5** for a system that will be used in production
  - **4** for a pre-production system
  - **3** for a test system

- **2** for a development system
- **1** for an experimental system

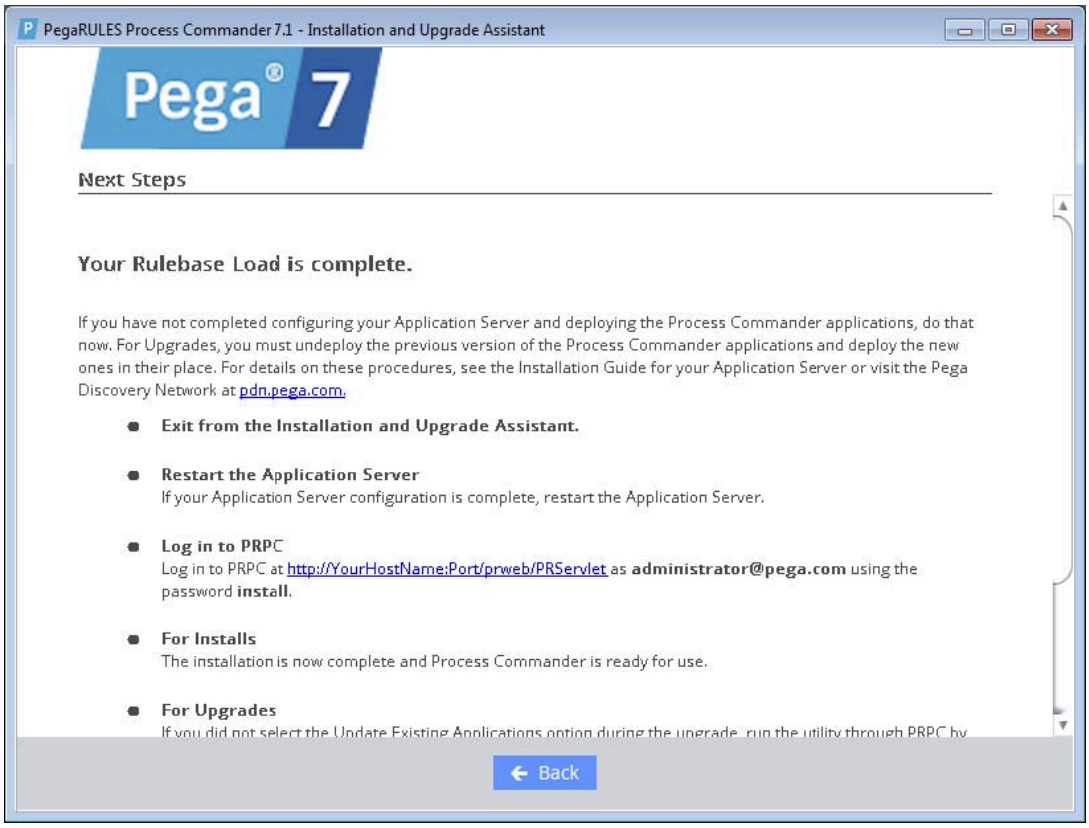
The Production Level can be updated from the App Explorer. Type Data-Admin-System in the search field and select SysAdmin > Class > Data-Admin-System to open your system.

12. Click **Start** to begin the Rulebase Load.



Installation logs automatically appear in this window. The log is also stored in the <distribution image>/scripts/logs directory.

The rulebase load time can last for several hours and will vary widely based on network proximity to the database server. The log window may appear inactive when the IUA is processing larger files. When the load is complete, the following screen opens.



If there is an issue, copy the log from the temporary directory to a safe location before exiting the IUA.

13. Click **Back** to return to the previous screen and then click **Exit** to close the IUA.

Installing from the command line

To use the **install.bat/.sh** script, first configure the **setupDatabase.properties** file.

Note: If no additional arguments are passed to the script, the script defaults to the values of the properties set in the **setupDatabase.properties** file. See Editing the setupDatabase.properties file.

1. If you have not done so already, edit the **setupDatabase.properties** file. See Editing the setupDatabase.properties file.
2. Open a command prompt and navigate to the **scripts** directory.
3. Type **install.bat** or **./install.sh** to run the script.

Installing the rulebase can take several hours, depending on the proximity of the database to the system running the install script. As the install proceeds, it reports performance metrics for each step of the installation process. When complete, you see a `BUILD SUCCESSFUL` message and the `Total Time` in minutes and seconds for the installation.

Application Server Setup

This section describes how to configure your WebSphere Application Server for use with Pega 7:

- Prerequisites
- Preparation
- Configuration
- Deploying Pega 7 Archives

Application server prerequisites

Before you start to set up your application server, confirm that the following prerequisites have been met:

- Your system includes a supported JDK. See the Platform Support Guide on the PDN for a list of supported versions. WebSphere provides a JDK.
- You have installed a supported version of the WebSphere application server. See the Platform Support Guide on the PDN for a list of supported versions.

Synchronizing hardware clocks

Pega 7 uses Hazelcast distributed clustering technology to share data and send events between server nodes. Like most clustering technology, Hazelcast assumes that all cooperating servers have synchronized clocks. Pega 7 validates time synchronization to ensure proper operations and triggers a PEGA0055 alert if clocks of sync between nodes. Pega 7 also reports these errors to AES. The impact of these errors can include timeout errors and other system problems.

Every operating system has a provision for referencing a common time standard like the one available at www.time.gov. On UNIX, this is the network time protocol daemon, ntpd. On Windows, you can set a similar service through the clock settings in the control panel or task bar. See the documentation for your specific hardware operating system for information about setting this critical service.

Configuring ports

Before you start to set up your application server, ensure that the following ports are open and available:

- Search (Elasticsearch): one TCP port in the range 9300-9399 (default 9300)
- Cluster communication - Hazelcast: multiple ports in the range 5701-5800 (default 5701)

Search

To enable communication between the host nodes in the cluster, open a TCP port in the range 9300-9399 on each node. (By default, Search uses port 9300.) These ports are used for internal node-to-node communication only, and should not be externally accessible.

Cluster Communication - Hazelcast

In a Pega 7 system, there can be multiple servers, or nodes, and each node can contain multiple JVMs. The number of available ports in this range needs to be greater than or equal to the greatest number of JVMs on any one node in the cluster. For example, if there are 3 JVMs on one node, and 7 JVMs on another node, there must be at least 7 ports available.

For cluster communication, the port range 5701-5800 needs to be left open. By default, the system begins with port 5701 and then looks for the next port in the sequence (5702, followed by 5703 and so on). To override the default port range, set the value of **cluster/hazelcast/ports** in the **prconfig.xml** file.

Configuring the Application Server

The section describes how to prepare and configure your application server.

The application server configuration process depends on two primary considerations.

1. Single versus Split-Schema
2. Base versus Admin User

Single versus Split Schema

In a split-schema configuration, Pega 7 uses the Java Naming and Directory Interface (JNDI) standard to identify and access the appropriate schema. One of the benefits of using JNDI is that it allows Pega 7 to access different schema while using only a single data source. The configuration of additional JNDI parameters is straightforward and covered in this section.

Base versus Admin User

Pega 7 minimally requires one JDBC data source. However, if you are using the recommended dual-user configuration, **Base + Admin user**, you must configure a second data source in the application server.

Enabling AWT to support Pega 7 Graphical Reports

The Abstract Window Toolkit (AWT) is Java's original platform-independent windowing, graphics and user interface widget toolkit. You must add a headless AWT setting to the JAVA options in the configuration file that is passed to Java at startup to allow Pega 7 to render and display graphics.

Include WebSphere Default Packages

Confirm that your installation of the WebSphere application server includes all the default packages.

In particular, you must include the package **EJBDeploy tool for pre-EJB 3.0 modules** which is required by Pega 7.

WebSphere profiles

Run the WebSphere Profile Management Tool to create a WebSphere Application Server profile in which to run Pega 7.

You can use either the **Typical** profile creation or the **Advanced** profile creation option. Both methods allow you to choose whether to apply Administrative Security to require users to log in with a password to start the Administrative Console.

- The **Typical** profile assigns default values for the Profile name, Node name, Host name, and access ports.

  On Windows platforms, the Typical option creates the application server as a Windows service by default. This means that you must manage the profile as a Windows service, for example when starting and stopping the server.

- The **Advanced** Profile lets you change the defaults. If you choose this option, you **must** choose to deploy the Administrative Console.

When you create the profile, record the following information for use later in the configuration.

- Profile name
- User name and password for the Administrative Console — if you choose to apply Administrative Security
- Port numbers for the profile, including:
  - Administrative Console Port
  - HTTP transport port
- Depending on your environment, you may also need to record the port numbers for the Administrative console secure port, the HTTPS transport port and the SOAP connector port. If you are unsure about these requirements, check with your application server administrator.

Using the WebSphere Administrative Console

Start the server for the new profile; launch the Administrative Console, and log in.

On Windows platforms, you can start the profile server and launch the Administrative Console from the Start menu. For example:

Start > All Programs > IBM WebSphere > IBM WebSphere Application Server V8.5 > Profiles > AppSvr01 > Start the server.

To launch the Administrative Console:

Start > All Programs > IBM WebSphere > IBM WebSphere Application Server V8.5 > Profiles > AppSvr01 > Administrative Console.

Setting JVM properties

Follow these steps to set the JVM properties.

1. In the Administrative window, on the left side of the screen, click Servers > Server Types > WebSphere application servers.
2. Select the server on which Pega 7 will run. The **Configuration** tab for the server opens.
3. In the **Server Infrastructure** section, expand **Java and Process Management** and click **Process Definition**. The Configuration tab opens.
4. In the **Additional Properties** section, click **Java Virtual Machine** to display its Configuration tab.
5. Select **Verbose garbage collection**.
6. Set the JVM memory options to increase the amount of system memory allocated to the application server running Pega 7:

   - Initial Heap Size (Xms): 4 GB for development, 8 GB for production
   - Maximum Heap Size (Xmx): 4 GB for development, 8 GB or larger for production, depending on your system configuration. See Application server memory requirements for more information.

   If your application server is using the recommended Oracle JVM, also add the PermSize and MaxPermSize settings.

   - PermSize (-XX:PermSize): 512 MB
   - MaxPermSize (-XX:MaxPermSize): 1024 MB

   Note: If the server does not have enough memory allocated to run Pega 7, the system may hang without an error message. Your values may need to be higher than these recommendations based on your server hardware and the number of other applications on the server.
7. In the **Generic JVM Arguments** field, enter the following argument to compress references to optimize heap space:

-Xcompressedrefs

8. In the **Generic JVM Arguments** field, enter these arguments:

   -Xverbosegclog:<*garbage log collection file name*>

9. In the **Generic JVM Arguments** field, enter the following to set the garbage collection policy to concurrent collector.

   -Xgcpolicy:gencon

   or

   -Xgcpolicy:optavgpause

10. In the **Generic JVM Arguments** field, enter the following argument to enable AWT for graphical reports:

    -Djava.awt.headless=true

11. Click **Apply**. A message appears at the top of the screen explaining that the changes were made.
12. Click **Save** in the confirmation message at the top of the page to save these changes to the master configuration.

Create URL providers

This procedure sets the required URL Pega reference and an explicit temporary directory for Pega 7. The temporary directory stores static data. It is important that the directory be properly specified and accessible to the Base user.

In WebSphere, specify this directory as a JNDI reference to a URL object.

1. In the WebSphere Administration console, select **Resources > URL > URL Providers** in the left frame.
2. Set the Scope level appropriate to your deployment, and click on the **Default URL Provider** link. The default URL configuration page opens.
3. Under the **Additional Properties** section, click the **URLs** link to display the URLs listing page.
4. Click **New** to display the Configuration page.
5. Complete this form as follows to define a URL for the NULL file that Pega 7 can use to discard erroneous error messages:

- In the **Name** field, enter **PRPCnone**
- In the **JNDI name** field, enter **url/pega/none**
- In the **Specification** field, enter
  - Windows: **file:///nul**
  - UNIX/Linux: **file:///dev/null**

6. Click **OK** and then **Save** in the confirmation message.
7. From the URLs page, click **New** again to return to the Configuration page.
8. Complete this form to create a URL specification for a temporary directory to store static data.
   - In the **Name** field, enter **PegaTempDir**
   - In the **JNDI** field, enter **url/initialization/explicittempdir**
   - In the **Specification** field, enter **file:///** followed by the full path to the temporary directory. Enter the path using forward slashes for both Windows and UNIX platforms. For example, on Windows the value might be: **file:///D:/Temp/PegaTempDir**.

   Note: The directory names you enter here are case sensitive. Be sure to enter the names of the directories exactly as they have been created on your system.

   Note: If the directory you specify does not exist, Pega 7 attempts to allocate it. It is good practice to allocate the directory on the system before specifying it here. The user that owns the Java process must have the appropriate permissions to use this directory, including write access. In particular, if you have J2 security enabled, ensure that this directory is accessible under your security policy.

   In a clustered deployment, each Pega 7 instance must have its own temporary directory. You cannot share a temporary directory with more than one Pega 7 instance.
9. Click **OK** and then **Save** in the confirmation message.

Creating a JDBC provider

1. In the Administrative console, click Resources > JDBC > JDBC Providers to display the JDBC Providers page.
2. In the **Scopes** dropdown, choose the scope level appropriate to your deployment.
3. Click **New** to display the JDBC provider wizard.
4. Complete the fields as follows:

- In the **Database type** dropdown, select Oracle .
- In the **Provider type** dropdown, choose Oracle JDBC Driver.
- In the **Implementation type** dropdown, select **Connection pool data source**.
- In the **Name and Description** fields, accept the defaults or enter a descriptive name and explanatory description for the driver you selected.

5. Click **Next**. The Enter database class path information screen opens.
6. In the **Directory** location field, confirm the entry, or enter the path to the driver class file or files listed in the class path field. See the *Platform Support Guide* for more information about supported drivers.

   The appropriate files depend on the Java version of your JVM:

   - For Java 6: ojdbc6.jar
   - For Java 7: ojdbc7.jar
   - For Java 8: ojdbc8.jar
   - For Java 6: sqljdbc4.jar
   - For Java 7: sqljdbc.jar
   - For Java 8: sqljdbc.jar

   Note: Be sure only the appropriate JAR file is loaded. Do not put both sqljdbc.jar and sqljdbc4.jar in the classpath.
7. Click **Next** to display the Summary screen.
8. Confirm that the settings are correct, and click **Finish** to return to the **JDBC Providers** page.
9. Click **Save** in the confirmation message.

Creating a data source

1. In the Administrative console, click **Resources > JDBC > JDBC Providers** to display the JDBC Providers page.
2. On the JDBC providers page, click on the name of the provider you just created to display the **General Properties** page.
3. Under the Additional Properties heading, click **Data Sources**.
4. Click **New** to display the data source wizard.

5. In the Step 1, **Enter basic datasource information** screen, enter the following information:
   - In the **Data Source name** field, enter **PegaRULES**
   - In the **JNDI name** field, enter **jdbc/PegaRULES**

     Tip: JNDI settings are case-sensitive.
6. Click **Next** to display the Step 2, **Enter database specific properties for the data source** screen.
7. Supply information about your Pega 7 database.

   Specify a URL for the Type 4 (thin client) JDBC driver
   jdbc:oracle:thin:@<*myServer*>:1521:<*myDatabase*>

   where *myServer* is the DB server name, *1521* is the port number it is using for communication, and *myDatabase* is the database name.

   In **Data store helper class name**, specify the Oracle data store helper.
8. Clear **Use this data source in container managed persistence (CMP)**.
9. Click **Next** to display the **Setup security aliases** page.
10. Click **Next** to display the Summary page.
11. On the Summary page, confirm that the settings are correct and click **Finish** to return to the JDBC Providers page.
12. Click **Save** in the confirmation message.
13. On the Data Sources page, click the **PegaRULES** link in the Name column to open the Configuration page for this data source. Then, under **Additional Properties**, click the **Custom Properties** link to display the Custom Properties page.
14. Click **New** to define any additional properties you may need for your database connection.

    After creating each property, click **OK** to save the property, and click **New** again to create the next property. Be sure to set the **Type** field appropriately for the **Value** of the connection property.

    Pega 7 requires the following properties; if these properties already exist, modify the value if needed:
    - Name: **webSphereDefaultIsolationLevel** Value: **2**
15. After you have set the necessary properties, click the **Save** link in the message at the top of the

page.

Defining database authentication credentials

1. In the link path at the top of the page, click **PegaRULES** to return to the PegaRules properties page. Then, under the **Related Items** section, click the link **JAAS – J2C authentication data**. You will define your database authentication credentials here.
2. Click **New** to specify the General Properties.
3. Complete this form as follows:
   - In the **Alias** field enter any name that uniquely identifies this J2C entry.
   - In the **User ID and Password** fields, enter the Base user name and password.
4. Click **OK** to return to the authentication data entries page, and click **Save** in the Messages section at the top of the page.
5. In the link path at the top of the page, again click **PegaRULES** to return to the PegaRules properties page.
6. In the **Security Settings** section near the bottom of the page, use the **Component-managed authentication alias** dropdown to select the J2C alias you just created.
7. Click **OK**, and then **Save** in the confirmation message on the Data sources page.
8. On the Data sources page, select the check box for PegaRULES and click **Test Connection** to confirm your data source configuration.

Configure WebSphere connection pool properties

At a minimum, set the Maximum data connections to 50.

Note: You should determine the best value of this setting based on your application architecture, usage profile and environment considerations. The database connection pool should be no smaller than the Work Manager pool.

To set the maximum connections:

1. In the WebSphere Administration console, open the Data sources page: **Resources > JDBC> Data sources**.
2. Click **PegaRULES**.
3. Under **Additional Properties**, click **Connection pool properties**.
4. Set the **Maximum connections** value to 50 or higher based on your environment needs. See the

PDN article *How to configure a non-blocking UI using Asynchronous Declare Pages* for more information about connections.

5. Click **Apply**.
6. Click **Save** in the Messages pane at the top of the screen to save the configuration changes.

Configuring WebSphere WorkManager

To deploy the PegaRULES archive, complete the following procedures to define a WorkManager. Pega 7 uses the WorkManager to run asynchronous tasks to support internal components such as agents, services, daemons, and child-requesters.

Complete the following procedures to define the WorkManager:

1. In the navigation menus on the left of the Administration Console expand **Resources**, expand **Asynchronous Beans**, and select **Work managers**
2. On the **Work managers** listing page, set the **Scope** to the level appropriate for your deployment, and click **New**.
3. Complete the **Configuration** page for the Work Manager.
    - In the **Name** field, enter a name to identify this Work Manager, for example, PegaWorkManager.
    - In the **JNDI name** field, enter the JNDI location **wm/PegaWorkManager**
    - In the **Service names** section, select **Security**.
    - In **Thread pool properties**, set **Maximum number of threads** to **20**.
    - Clear the **Growable** check box.
4. Click **OK**.
5. In the **Messages** box at the top of the page, click **Save**.

Set HTTP transport channel custom property

To support the ability to open files attached to work objects in Pega 7, set custom property **CookiesConfigureNoCache** to false on the transport chain in the Web Container settings for the Pega 7 server.

Note: For earlier versions of WebSphere, custom property **CookiesConfigureNoCache** was set to false by default so that cookies could be cached. The current version of WebSphere has this property set

to true by default. Because Pega 7 requires this caching to allow users to open attached files directly from a work object, you must change the property setting.

1. In the Administrative console, select Servers > Server Types > WebSphere application servers to display the Application servers page.
2. Click the name of your Pega 7 server to display the Configuration page.
3. Under Container Settings, expand **Web Container Settings**, and then click **Web Container transport chains**.

   This console displays the listing page for transport chains.
4. Click the name of the appropriate transport chain.

   - If your site is using the default transport configuration for this server, select WCInboundDefault.
   - If you have enabled SSL for the Pega 7 port, WCInboundDefaultSecure.
   - If your site has defined a custom transport chain for this server, select that chain name.

   The console displays the Configuration page for the selected transport chain.
5. Under Transport Channels on this page, click **HTTP inbound channel**.

   The console displays the Configuration page for the selected transport channel.
6. Under Additional Properties, click **Custom Properties**.

   The console displays the listing page for this channel's custom properties.
7. Click **New**

   The console displays the form for a new custom property.
8. Complete this form.

   - In the **Name** field, enter **CookiesConfigureNoCache**
   - In the **Value** field, enter **False**
9. Click **OK**, and then **Save** in the Messages box on the top of the Custom Properties listing page.

Configure pass-by-reference property

To improve performance, enable pass by reference in the WebServer application server Object Request Broker services.

1. In the Administrative console, select Servers > Server Types > WebSphere application servers to display the Application servers page.
2. Click the name of your Pega 7 server to display the Configuration page.
3. Under Container Settings, expand **Container Services**, and then click **ORB service**.
4. Select **Pass by Reference**.
5. Click **OK**.
6. Click **Save** in the Messages box at the top of the Application servers page to save the setting.

Creating JNDI binding identifiers

Modify the JDNI settings so the data source uses the correct schemas. Create these binding identifiers:

- One that points to the **PegaRULES** default schema
- One that points to the **PegaDATA** default schema
- Optional: one that points to **PegaRULESdbAdmin**
- Optional: one that points to database/databases/PegaDATA/**dataSourceAdmin**

To create new Binding Identifiers:

1. In the Administrative console, select Environment > Naming > Name Space Bindings to display the Name space bindings page.
2. Select the correct **Scope** for your environment and click **New**.
3. For the binding type, select **String** and click **Next**.
4. On the Step 2: **Specify basic properties** screen, enter the following values:
    - Binding identifier: **PegaRULESDefaultSchema**
    - Name in the name space relative to lookup name prefix: **prconfig/database/databases/PegaRULES/defaultSchema**
    - String Value: the schema name of your Pega Database Rules Schema.
5. Click **Next**.
6. On the **Summary** panel, click **Finish**.
7. Click **Save** in the Messages box at the top of the **Name Space Bindings** screen.
8. Repeat steps 2 -6 for the Data Schema except: On the **Step 2: Specify basic properties** screen,

enter the following values for the Data Schema.

- Binding identifier: **PegaDATADefaultSchema**
- Name in the name space relative to lookup name prefix: **prconfig/database/databases/PegaDATA/defaultSchema**
- String Value: the schema name of your Pega Database Data Schema

9. Click **Save** in the Messages box at the top of the **Name Space Bindings** screen.

Deploying Pega 7 application archives

This section describes how to deploy the Pega 7 application archives after you configure your WebSphere application server. This includes:

Pega 7 application	File name
PegaRULES	prweb.war or prpc_j2ee14_ws.ear
System Management	prsysmgmt.war
Help	prhelp.war

Note: When you restart the server following deployment of the archives, the first node you bring up will become the default search node.

The PegaRULES application archive is available both as an EAR file, **prpc_j2ee14_ws.ear** and a WAR file, **prweb.war**. Pegasystems recommends deploying the EAR file for all new deployments. You can successfully deploy the Pega 7 application WAR file on the application server but if you use the EAR file, you can add advanced J2EE features as your application grows or when you migrate to a production environment.

Caution: Do not install both prweb.war and prpc_j2ee14_ws.ear.

Deploying the Pega 7 applications

WebSphere automatically starts the application, prweb.war or prpc_j2ee14_ws.ear, when it is deployed. When the application starts , you may see error messages for missing resources and references. Ignore these messages. You supply these resources as you complete the installation process. Be sure to **stop the application after deploying**.

To deploy the Pega 7 application using the prpc_j2ee14_ws.ear file included in your distribution media:

1. Make sure the WebSphere Application Server is running. Log into the Administration console.
2. From the left frame, select Applications > New Application.
3. Click **New Enterprise Application**.

4. Click the **Browse** button and select **prpc_j2ee14_ws.ear** from the archives directory on your installation media. Click **Open** and then click **Next**.

   WebSphere displays Preparing for the application installation.

5. Select **Detailed - Show me all installation options and parameters**.

   This option allows you to review all the installation options for the application, including the default bindings and resource mappings.

6. Click **+** to expand **Choose to generate default bindings and mappings.**

7. Complete this page.

   - Check **Generate Default Bindings.**
   - Check **Use default virtual host name for Web and SIP modules**.
   - Leave the other default settings and click **Next.**

   WebSphere displays Application Security Warnings.

8. Scroll to the bottom on this page and click **Continue** to display a wizard where you can specify installation options.

   This security file allows Pega 7 to run when J2EE Security Checking is enabled.

   This section of the installation process is a series of steps under the general heading of **Install New Application**.

9. For Step One, accept the defaults and click **Next.**

10. Continue through the next steps, either accepting the defaults, or customizing for your organization, as needed.

11. Locate the step where you **Map resource references to resources**.

12. In the **Map resource references to resources** step, there are three rows that include "explicittempdir" in the Resource Reference column. Use the find tool on your browser to find the correct rows for:

    - EJB EngineCMT bean
    - EngineBMT beans
    - prweb.war module

13. For each of the three rows, change the value in the **Target Resource JNDI Name** field to the temp directory, for example **url/initialization/explicittempdir**. This maps the location you

specified in the URL provider you created to the corresponding Resource Reference in the application, so that the application will use the location for the **PegaTempDir**. Use the Browse button and Apply to change each of the three values.

14. Click **Next**

    Depending on your configuration, you may see a set of warnings related to missing resource references. These warnings are informational. Review the warnings and then continue.

    Note: These are resource references that are defined in web.xml, the deployment configuration files for the application, but not mapped to resource definitions in your application. In the page, **Map resources to references**, they are mapped to the Target Resource JNDI Name **url/pega/none**, indicating that they are not used. Pegasystems provides these references for J2EE compliance, but their use is optional. You can continue with the deployment.

15. At the bottom of the Warnings page, click **Continue**.

16. Click **Next** as needed to continue through the remaining steps, accepting the defaults, or setting them to the requirements of your organization.

17. On the **Summary** page, click **Finish**.

    The system begins installing the EAR file, which can take a few minutes. When the installation completes successfully, WebSphere displays a success message similar to the following: "Application Pega 7 installed successfully."

18. Click **Save directly to the master configuration**.

Deploying the System Management Application and on-line help

In addition to the Pega 7 application, you must also deploy the online help application — **prhelp.war** and the System Management application — **prsysmgmt.war**.

For more details on the System Management Application, refer to the *System Management Application Reference Guide* on the PDN.

Complete the following steps to deploy prsysmgmt.war and prhelp.war.

1. In the Preparing for the application installation screen, select Local file system and click Browse to select the prsysmgmt file.

   If you do not see the Preparing for the application installation screen, from the left frame, select Applications > New Application.

2. Click **Browse** and navigate to select the application war file, either prsysmgmt.war or prsysmgmt.ear from the archive directory in your installation media.

3. Click **Open** and then click **Next** to begin the installation.

4. Click **Detailed - Show all installation options and parameters**.
5. Expand **Choose to generate default bindings and mappings**.
6. Select the **Generate Default Bindings** check box, and leave the other settings at their defaults.
7. Click **Next**.

   You may see some security warnings.
8. Click **Continue** to bypass the warnings.

   WebSphere displays the **Install New Application** wizard.
9. Accept the defaults and click **Next** until you get to the **Map context roots for Web Modules** step.
10. In the **Map context roots for Web Modules** step, enter **prsysmgmt** as the context root, and click **Next**.
11. Accept the defaults and click **Next** on the remaining steps.
12. On the Summary page, click **Finish**.

    WebSphere displays a message, **Installing . . .**, and updates it with information during the install. When the installation is complete, you see the following message:
13. Click the **Save** link to save the changes to the master configuration and return to the first page of the installation series.
14. Repeat steps 2 - 13 to deploy prhelp.war using the same procedure as prsysmgmt.war.

    Use the name of the file, **prhelp**, as the context root and deploy to the same server.

Assigning the PegaDiagnosticUser Role to your System Management Application users

Users accessing the System Management Application must be assigned the role PegaDiagnosticUser. If the user does not have this role they will be challenged for authentication when they attempt to access the System Management Application.

You can use the System Management Application to monitor and control caches, agents, requestors, listeners, and other processing. For more information, see the *System Management Application Reference Guide* on the PDN.

The default deployment configuration assigns the PegaDiagnosticUser role to administrative users. To restrict access to the System Management Application, use your application server tools to assign the

PegaDiagnosticUser role to any other users that must be able to access the System Management Application.

The role PegaDiagnosticUser is associated with the System Management Application through a security-constraint element on the Diagnostic Data servlet defined in the prweb.war application's web.xml file. In EAR file deployments, the web.xml file is located in the prweb.war file packaged inside the prpc_*.ear file:

```
<security-constraint>
 <web-resource-collection>
  <web-resource-name>Diagnostic Data</web-resource-name>
  <description>Serves diagnostic files generated by the JMXclient</description>
  <url-pattern>/DiagnosticData</url-pattern>
  <http-method>GET</http-method>
  <http-method>POST</http-method>
 </web-resource-collection>
 <auth-constraint>
  <role-name>PegaDiagnosticUser</role-name>
 </auth-constraint>
</security-constraint>
```

If you do not want to restrict access to the System Management Application, edit the web.xml file and delete the <security constraint> element.

Changing this setting in the web.xml file may require redeploying the PegaRULES Application.

Note: If you want to allow limited access to the SMA, use the Map Special Subjects dropdown to select All Authenticated in Application's Realm, or click **Map Users** to assign the role to specific users.

To assign the role:

1. Select Applications > Application Types > WebSphere enterprise applications.
2. Click the name of your Pega 7 core application, for example **prpc_j2ee14_ws.ear**, to open the Configuration page.
3. In the **Detail Properties** section, click **Security role to user/group mapping**.
    - If you installed the EAR archive, prpc_j2ee14_ws.ear, the **PegaDiagnostic** role is set to **Everyone**:
    - If you installed the WAR archive, prweb.war, the **PegaDiagnosticUser** role is set to **None.**
4. Select **PegaDiagnosticUser** and set the proper access level for your site.

- If you do not want to allow access to the SMA, use the **Map Special Subjects** dropdown to select **None**.
- If you want to allow limited access to the SMA, use the **Map Special Subjects** dropdown to select **All Authenticated in Application's Realm**, or click **Map Users . . .** to assign the role to specific users.
- If you want to allow open access to the SMA, use the **Map Special Subjects** dropdown to select **Everyone**.

Post Deployment Configuration

This section describes the post deployment activities that are performed in the system once you have completed the setup and configuration of your WebSphere application server and deployed the Pega 7 archives.

- Starting applications
- Logging in into Pega 7
- Enable Help and System Administration applications
- Configure Directed Web Access
- Configure search index host node settings
- Change the administrator password
- Install Pega-supplied applications

Starting the Pega 7 applications

Ensure that the application server is running and start the prsysmgmt, prweb, and Pega 7 applications.

To start the application:

1. In the WebSphere Administrative Console, select Applications > Application Types > WebSphere enterprise applications.
2. Select the three applications and click **Start**.

   The application status turns green and the system displays a success message.

Logging in to Pega 7

To test the installation, log into the Pega 7 web application, replacing the <server> and <port> values with your specific values.

http://<server>:<port>/prweb/PRServlet

Use the following credentials to log in:

- **User ID:** administrator@pega.com
- **Password:** install

After logging in, Pega 7 automatically launches a process to index the rules in the system to support full-text search. While this process is running, you may notice some delay in the responsiveness of the Pega

7 user interface. This process usually takes from 10 to 15 minutes to complete depending on your system configuration.

If the index ends without generating an error message, the installation is successful.

Enabling help and System Management applications

The Help and System Management web applications are accessed from the Pega 7 interface through a URL. Either application can be deployed to a different application server than the core Pega 7 application.

To enable these applications:

1. From the **Designer Studio**, select the System > Settings > URLs menu item.
2. In the **Online Help URL** field, enter the URL for the Help application in the format:

   `http://<host>:<port>/prhelp`
3. In the **SMA URL** field, enter the URL for the System Management Application in the format:

   `http://<host>:<port>/prsysmgmt/getnodes.action`
4. Click **Save**.

Note: You must log out of Pega 7 and log back in for these changes to take effect.

Configuring Directed Web Access

Directed Web Access (DWA) allows external users that do not have a Pega 7 Operator ID to process an assignment in your application on a one-time basis. The URL you specify here is provided in an email to the external user to access Pega 7 and may identify a proxy server.

To configure DWA:

1. Select the **Designer Studio** System > Settings > URLs menu item.
2. In the **Public Link URL** field, enter the URL that you want to provide in emails in this format:

   http://<host>:<port>/prweb
3. Click **Save**.

Note: You must log out of Pega 7 and log back in for these changes to take effect.

Configuring search index host node settings

Pega 7 supports full-text search for rules, data instances, and work objects. By default, search indexing is enabled and indexing starts automatically when you start the application server after installing Pega 7. The first node that starts after installation becomes the default initial search node. The default index directory is PegaSearchIndex in your temporary directory.

After the Search indexes are completely built, you can change the default settings. Do not stop or bring down the default node until the search indexes build completely. The Search Landing Page displays the status.

1. Check your directory sizes:
    - Ensure that the directories for all Elasticsearch host nodes have sufficient free space to hold the Elasticsearch indexes.
2. Open the **Designer Studio** > System > Settings> Search landing page and expand Search Index Host Node Setting.
3. Specify one node to set as the Host Node. If necessary, delete all but one node. This is the node on which Elasticsearch indexes will be built.

    Note: Do not include more than one node in the Search Index Host Node Setting list. Including more than one node in the list at this point may cause duplicate index builds and compromise system performance. You will create additional nodes later in this process.

4. Verify the Search Index Host Node ID and the Search Index File Directory.
5. Expand Automated Search Alerts, and enable **Automatically Monitor Files**.
6. Click **Submit** to save the settings.
7. After the first indexing is complete, add additional Host Nodes. The system automatically replicates the indexes on the new nodes.

    Note: You must configure a minimum of 2 Elasticsearch host nodes. Pegasystems recommends that you configure a minimum of 3 nodes for maximum fault tolerance. You may need more than 3 nodes depending on the size of your cluster.

8. To enable communication between Elasticsearch host nodes in the cluster, open a TCP port in the range 9300-9399 on each node. (By default, Elasticsearch uses port 9300.) These ports are used for internal node-to-node communication only, and should not be externally accessible.

Changing the administrator password

Because the default password for the system administrator is widely published, it is important to protect your system after an installation by changing the password for the Adminstrator@pega.com operator ID.

To change the password:

1. From the Operator Menu located to the right of the Designer Studio header, select the **Profile** option.
2. Click **Change Password**.
3. Verify the **Current Password**, then enter and confirm the **New Password**.
4. Click **Save**.

Installing Pega-supplied applications

Install any Pega-supplied applications now. Follow the instructions in the *Installation Guide* provided on the distribution media for the Pega-supplied applications.

Caution: If you plan to install a Pega-supplied applications on top of Pega 7.1.8, you must grant the database user permissions, including SELECT_CATALOG_ROLE as described in Database users. Some Pega-supplied applications use triggers and the user will need the permissions to determine if Pega 7 will need to drop triggers that reference the updates cache and rule view tables. During startup, Pega 7 checks for triggers that reference the updates cache and rule view tables; if these triggers exist, Pega 7 attempts to drop them. If the Base or Admin user does not have the correct permissions, Pega 7 cannot drop the triggers and fails to start up. If you installed the Pega-supplied application before you deployed Pega 7 , the deployment process automatically drops the triggers and this error does not occur

Appendix A: setupDatabase.properties

This appendix contains an example of how the setupDatabase.properties file should look when the relevant properties for your installation have been entered.

```
# Properties File for use with PRPC Utilities. Update this file

# before using setupDatabase.bat/sh script.

# Set the DB connection

################### COMMON PROPERTIES - DB CONNECTION (REQUIRED)
###################

#########################################################################
####################

# For database that uses multiple JDBC driver files (such as DB2). You may specify

# semicolon separated values for 'pega.jdbc.driver.jar'

#

# pega.jdbc.driver.jar -- path to jdbc jar

#

# pega.jdbc.driver.class -- jdbc class. valid values are:

#

# Oracle 10g (Type 4) oracle.jdbc.OracleDriver

# IBM DB/2 Type 4 com.ibm.db2.jcc.DB2Driver

# SQL Server 2008 com.microsoft.sqlserver.jdbc.SQLServerDriver

# PostgreSQL 9.1.4 org.postgresql.Driver

#

# pega.database.type valid values are: mssql, oracledate, udb, db2zos, postgres

#

# pega.jdbc.url valid values are:

#

# Oracle 10g (Type 4) jdbc:oracle:thin:@//localhost:1521/dbName
```

IBM DB/2 z / OS jdbc:db2://localhost:50000/dbName

# IBM DB/2 Type 4
jdbc:db2://localhost:50000/dbName:fullyMaterializeLobData=true;fullyMaterializeInputStreams=true;progressiveStreaming=2;useJDBC4ColumnNameAndLabelSemantics=2;

SQL Server 2008
jdbc:sqlserver://localhost:1433;selectMethod=cursor;sendStringParametersAsUnicode=false

PostgreSQL 9.1.4 jdbc:postgresql://localhost:5432/dbName

#

pega.jdbc.username db username

pega.jdbc.password db password

#

** If you are installing or upgrading on db2 or udb you should do the following:

set the rules.schema.name to the schema name required

update the udb.conf file in the /config directory if you have additional

connection properties that need to be set. (schema name will be automatically added)

pega.jdbc.driver.jar=/opt/tomcat/lib/postgresql.jdbc4.jar

pega.jdbc.driver.class=org.postgresql.Driver

pega.database.type=postgres

pega.jdbc.url=jdbc:postgresql://localhost:5432/pega

pega.jdbc.username=pega

pega.jdbc.password=pega

#Uncomment this property and add a list of ; delimited connections properties

#For example jdbc.custom.connection.properties=user=usr;password=pwd

#jdbc.custom.connection.properties=

#Rules schema name : Used for all databases.

#The user name is used for default schema name

rules.schema.name=pegarules

```
# Data schema name : Used for systems running on a Split Schema
# The value of rules.schema.name is the default value for data.schema.name
data.schema.name=pegadata
#User Temp Directory. Will use default if not set to valid directory
user.temp.dir=
#z/OS Site Specific Properties file
pega.zos.properties=
# Generate schema will be skipped if this property is set to true
# Note: Leave this property blank if you need to generate the schema
bypass.pega.schema
# Generate UDF will be skipped if this property is set to true
# Note: Leave this property blank if you need to generate the UDF
bypass.udf.generation
# The utility will skip disabling and enabling of Pega Triggers during processing
# if this property is set to true
# Note: Leaving this property blank allows the utility to control the Pega Triggers.
bypass.trigger.control
# Note: The utility will skip any DDL Generation if all three bypass properties above
# are set to true.
# Truncate UpdatesCache will be skipped if this property is set to true
# Note: This property should be left blank so that the UpdatesCache table
# may be truncated automatically by the installer. If your site
# chooses to bypass this reset of the UpdateCache table, then the
# site must do the truncate of the 'PR_SYS_UPDATESCACHE' in the Data
# schema, manually after the install, upgrade or update before the
# system is started up.
```

#bypass.db2zos.truncatecache

Run assembler after install/upgrade? : Defaults to false.

This setting is to generate the Assembled Java classes

run.assembler=

Load Assembled Java Classes

import.assembled.classes=true

Rebuild Database Rules Indexes after Rules Load to improve Database Access Performance

It can be executed manually by running the stored procedure SPPR_REBUILD_INDEXES

Default is false except for z/OS, where the default is true

rebuild.indexes

The system name uniquely identifies a single Process Commander System.

Since multiple PRPC Systems may reference the same database, it is important that each

system has a unique name in order to distinguish them from each other.

During installs, the following system name will be created.

system.name=pega

During installs, the above system name is generated with the following production level.

The system production level can be set to one of the below integer values (1-5):

5 = production;

4 = preproduction;

3 = test;

2 = development;

1 = experimental

production.level=2

Is this a multitenant system?

A multitenant system allows organizations to act as distinct PRPC installations

multitenant.system=false

Run Update Existing Applications activity after upgrade?

Default setting is false.

For pre-7.1 upgrades: Update Existing Applications is always run, regardless of this setting.

For 7.1+ upgrades or updates: Upgrade Existing Applications is run if this setting is set to true.

Update Existing Applications can be run from upgrade scripts, prpcUtils, or by directly launching in PRPC after upgrade.

update.existing.applications=false

codeset and version for loading hotfixes

import.codeset.name=Customer

import.codeset.version=06-01-01

Pre build the Pega Conclusions.

build.conclusions=

Workload manager to load UDFs into db2zos

db2zos.udf.wlm

Run RuleSet Cleanup

Generate and execute a SQL script to clean old rulesets and their rules from the system

If you would like to only generate the script, not execute it, see cleanup.bat or cleanup.sh script

run.ruleset.cleanup

uncomment to generate timing benchmarks

generate.benchmarks=true

benchmark.warning.threshold.value=

benchmark.kill.threshold.value=

#Schema file name to be used for reversal

#reversal.schema.file.name=